PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-069191
(43)Date of publication of application : 03.03.2000

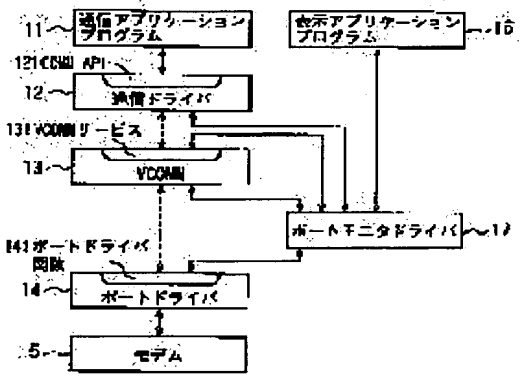
(51)Int.CI. H04M 11/00
G06F 3/00
G06F 13/00

(21)Application number : 10-234351
(22)Date of filing : 20.08.1998

(71)Applicant : TOSHIBA CORP
(72)Inventor : ISHIKAWA KATSUTOSHI

(54) COMPUTER SYSTEM AND DATA COMMUNICATION INFORMATION DISPLAY METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a computer system capable of displaying a variety of data communication information such as the disconnection reason of a line, a connecting destination telephone number and charging information without accompanying the improvement of a communication application program.
SOLUTION: A communication driver 12 communicates with a VCOMM 13 by using a VCOMM service 131, the VCOMM 13 communicates with a port driver 14 by using a port driver function 141 and a port monitor driver 17 tentatively hooks (snatches or steals) the utilization of the VCOMM service 131 of the communication driver 12 and the utilization of the port driver function 141 of the VCOMM 13. Then, a display application program 16 displays the variety of the data communication information (disconnection reason of the line, the connecting destination telephone number and the charging information, etc.), obtained by the hook of the port monitor driver 17 to a user.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2000－69191
（P2000－69191A）
(43)公開日　平成12年３月３日(2000.3.3)

(51)Int.Cl.'		識別記号	FI			テーマコード'(参考)
Ｈ０４Ｍ	11/00	３０２	Ｈ０４Ｍ	11/00	３０２	５Ｂ０８９
Ｇ０６Ｆ	3/00	６５２	Ｇ０６Ｆ	3/00	６５２Ａ	５Ｋ１０１
	13/00	３５３		13/00	３５３Ｕ	

審査請求　未請求　請求項の数10　ＯＬ　（全 10 頁）

(21)出願番号　特願平10－234351

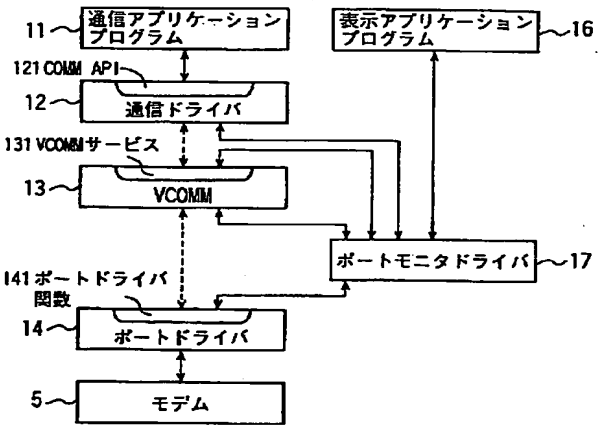
(22)出願日　平成10年８月20日(1998.8.20)

(71)出願人　000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地
(72)発明者　石川　勝敏
東京都青梅市末広町２丁目９番地　株式会社東芝青梅工場内
(74)代理人　100058479
弁理士　鈴江　武彦　（外６名）

Ｆターム(参考)　5B089 GA05 GA35 GB01 JB10 JB15
KA13 LB14
5K101 KK15 KK16 NN18

(54)【発明の名称】　コンピュータシステムおよびデータ通信情報表示方法

(57)【要約】
【課題】回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報の表示を通信アプリケーションプログラムの改良を伴なわずに行なうことを可能とするコンピュータシステムを提供する。
【解決手段】通信ドライバ１２は、ＶＣＯＭＭサービス１３１を用いてＶＣＯＭＭ１３と通信し、また、ＶＣＯＭＭ１３は、ポートドライバ関数１４１を用いてポートドライバ１４と通信するものであり、ポートモニタドライバ１７は、この通信ドライバ１２のＶＣＯＭＭサービス１３１の利用とＶＣＯＭＭ１３のポートドライバ関数１４１の利用とを一時的にフック（横取り、あるいは盗み取り）する。そして、表示アプリケーションプログラム１６は、このポートモニタドライバ１７のフックにより得られる各種データ通信情報（回線の切断理由、接続先電話番号および課金情報など）をユーザに対して表示する。



【特許請求の範囲】
【請求項１】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記デバイスに対するＩ／Ｏコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、
前記通信ポートを駆動制御するポートドライバと、
前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたＩ／Ｏコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、
前記Ｉ／Ｏコマンドに対する応答を前記通信制御手段より奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記応答から所定のデータ通信情報を抽出するデータ通信情報抽出手段と、
前記データ通信情報抽出手段により抽出されたデータ通信情報を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。
【請求項２】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記デバイスに対するＩ／Ｏコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、
前記通信ポートを駆動制御するポートドライバと、
前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたＩ／Ｏコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、
前記Ｉ／Ｏコマンドに対する応答を前記通信制御手段より奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記応答から回線の切断理由を抽出するデータ通信情報抽出手段と、
前記データ通信情報抽出手段により抽出された回線の切断理由を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。
【請求項３】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記デバイスに対するＩ／Ｏコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、
前記通信ポートを駆動制御するポートドライバと、
前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたＩ／Ｏコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、
前記Ｉ／Ｏコマンドに対する応答を前記通信制御手段より奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記応答から接続先電話番号を抽出するデータ通信情報抽出手段と、
前記データ通信情報抽出手段により抽出された接続先電話番号を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。
【請求項４】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記デバイスに対するＩ／Ｏコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、
前記通信ポートを駆動制御するポートドライバと、
前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたＩ／Ｏコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、
前記Ｉ／Ｏコマンドに対する応答を前記通信制御手段より奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記応答から課金情報を抽出するデータ通信情報抽出手段と、
前記データ通信情報抽出手段により抽出された課金情報を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。
【請求項５】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記通信ポートを駆動制御するポートドライバと、
前記デバイスに対して発行されるＩ／Ｏコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、
前記Ｉ／Ｏコマンドに対する応答を前記通信制御手段より奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記応答から所定のデータ通信情報を抽出して要求元に引き渡すデータ通信情報抽出手段とを具備することを特徴とするコンピュータシステム。
【請求項６】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記デバイスに対して発行されるＩ／Ｏコマンドおよびその応答をその発行元と前記デバイスとの間で奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記Ｉ／Ｏコマンドおよびその応答を本来の受け取り先に返却するＩ／Ｏコマンドリストア手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記応答から所定のデータ通信情報を抽出して要求元に引き渡すデータ通信情報抽出手段とを具備することを特徴とするコンピュータシステム。
【請求項７】　回線網との接続に用いられるデバイスに

対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムのデータ通信情報表示方法であって、
前記デバイスに対して発行されるＩ／Ｏコマンドおよびその応答をその発行元と前記デバイスとの間で奪取するステップと、
前記奪取した前記応答の内容から所定のデータ通信情報を抽出するステップと、
前記抽出したデータ通信情報を表示するステップとを具備することを特徴とするデータ通信情報表示方法。
【請求項８】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記デバイスに対するＩ／Ｏコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、
前記通信ポートを駆動制御するポートドライバと、
前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたＩ／Ｏコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、
前記Ｉ／Ｏコマンドおよびその応答を前記通信制御手段より奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記Ｉ／Ｏコマンドおよびその応答を前記通信制御手段に返却するＩ／Ｏコマンドリストア手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記Ｉ／Ｏコマンドまたはその応答から回線の切断理由を抽出するデータ通信情報抽出手段と、
前記データ通信情報抽出手段により抽出された回線の切断理由を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。
【請求項９】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記デバイスに対するＩ／Ｏコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、
前記通信ポートを駆動制御するポートドライバと、
前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたＩ／Ｏコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、
前記Ｉ／Ｏコマンドおよびその応答を前記通信制御手段より奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記Ｉ／Ｏコマンドおよびその応答を前記通信制御手段に返却するＩ／Ｏコマンドリストア手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記Ｉ／Ｏコマンドまたはその応答から接続先電話番号を抽出するデータ通信情報抽出手段と、
前記データ通信情報抽出手段により抽出された接続先電話番号を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。
【請求項１０】　回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポートを備えたコンピュータシステムにおいて、
前記デバイスに対するＩ／Ｏコマンドの発行により他のコンピュータシステムとのデータ通信を実行するデータ通信手段と、
前記通信ポートを駆動制御するポートドライバと、
前記データ通信手段と前記ポートドライバとの間に介在し、前記データ通信手段により発行されたＩ／Ｏコマンドに応じて前記ポートドライバを動作制御する通信制御手段と、
前記Ｉ／Ｏコマンドおよびその応答を前記通信制御手段より奪取するＩ／Ｏコマンドフック手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記Ｉ／Ｏコマンドおよびその応答を前記通信制御手段に返却するＩ／Ｏコマンドリストア手段と、
前記Ｉ／Ｏコマンドフック手段が奪取した前記Ｉ／Ｏコマンドおよびその応答から課金情報を抽出するデータ通信情報抽出手段と、
前記データ通信情報抽出手段により抽出された課金情報を表示するデータ通信情報表示手段とを具備することを特徴とするコンピュータシステム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】この発明は、たとえばモデムを用いたデータ通信を実行する通信アプリケーションプログラムが動作するコンピュータシステムおよび同システムのデータ通信情報表示方法に係り、特に、回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報の表示を通信アプリケーションプログラムの改良を伴なわずに行なうことを可能とするコンピュータシステムおよびデータ通信情報表示方法に関する。
【０００２】
【従来の技術】近年、デスクトップタイプやノートブックタイプなどと称される様々な種類の個人向けコンピュータ（パーソナルコンピュータ）が開発されている。また、この種のパーソナルコンピュータでは、インターネットや電子メールなどの普及に伴なって、たとえばモデムなどといった、回線網との接続に用いられるデバイスに対してコマンド信号を送出するための通信ポート（あるいは、そのデバイスそのもの）を内蔵するのが一般的である。
【０００３】ところで、この種のパーソナルコンピュータ上で動作する、たとえばモデムを用いたデータ通信を実行する通信アプリケーションプログラム（インターネットブラウザソフトウェアや電子メールソフトウェアな

ど）は、たとえば文献「Ｗｉｎｄｏｗｓ９５システムプログラム開発」（Ｗａｌｔｅｒ　Ｏｎｅｙ著、太田博志、小松克行訳、アスキー出版局出版、１９９７年１０月１１日　初版発行）などに記載された所定の手順でデータ通信を行なっているが、接続先電話番号や通信回線が切断された際の切断理由、あるいは課金情報などの各種データ通信情報を、モデムからのエコーバックやＡＴコマンドなどから取得したり、その取得したデータ通信情報をユーザインターフェースとして表示するといったことは行なっていなかった。すなわち、たとえモデムがデータ通信終了時にＡＴコマンドのエコーバックとして回線の切断理由を示すデータ通信情報を出力しても、通信アプリケーションプログラムはそれらを無視していた。

【０００４】

【発明が解決しようとする課題】この通信アプリケーションプログラムによるデータ通信情報の無視は、データ通信情報そのものが、安定度の高い有線回線を利用したデータ通信の実行時にはさほど必要視されていなかったことによる。しかしながら、最近では、携帯電話やＰＨＳなどの無線回線を利用したデータ通信も盛んに行なわれるようになってきており、ユーザの意図しない回線の切断が大幅に増加している。そして、回線の切断理由を得られないユーザは、このような場合、そのほとんどが電波状況の悪化によって回線が切断されたと考え、改めて回線を接続しようとするが、電波状況の悪化によって回線が切断されたのではなく、接続先の着信拒否により回線が切断されたときには、ユーザに無駄な操作を強いることになってしまう。

【０００５】この発明はこのような実情に鑑みてなされたものであり、回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報の表示を通信アプリケーションプログラムの改良などを伴なわずに行なうことを可能とするコンピュータシステムおよびデータ通信情報表示方法を提供することを目的とする。

【０００６】

【課題を解決するための手段】この発明は、前述した目的を達成するために、たとえばインターネットブラウザソフトウェアや電子メールソフトウェアなどがモデムなどに対して発行するＩ／Ｏコマンドおよびその応答を、その発行元とモデムとの間のいずれかで盗み取り、この盗み取ったＩ／Ｏコマンドおよびその応答から回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報を抽出するとともに、その盗み取ったＩ／Ｏコマンドおよびその応答を、本来の受け取り先に何もなかったかのように戻すようにしたものである。

【０００７】この発明においては、従来のシステム資源にＩ／Ｏコマンドおよびその応答の奪取をなんら認識させることがないため、通信アプリケーションプログラムの改良などを伴なわずに各種データ通信情報を表示することを可能とし、たとえば回線の切断理由を表示すれば、接続先の着信拒否により回線が切断されたときなど、ユーザに無駄な操作を強いることを防止する。

【０００８】また、回線の接続時に接続先電話番号を抽出して表示すれば、ユーザに接続先の電話番号が正しいかどうか確認させる機会を与えることができ、誤った電話番号を指定して回線の接続を実施していたときなど、その誤りをより早く認識させることが可能となる。

【０００９】さらに、回線の切断時に課金情報を抽出して表示すれば、たとえばシステム本体に備えられたタイマ機能により回線接続時間を計時して課金額を算出するといった複雑かつ誤差を生じ易い処理を行なわずとも、容易かつ正確に課金額をユーザに提供することが可能となる。

【００１０】

【発明の実施の形態】以下、図面を参照してこの発明の実施形態を説明する。図１は、この発明の実施形態に係るコンピュータシステムの概略機器構成を示す図である。この実施形態のコンピュータシステムは、たとえばデスクトップやノートブックタイプのパーソナルコンピュータなどであり、図１に示すように、ＣＰＵ１、システムメモリ２、ディスプレイコントローラ３、磁気ディスク装置（ＨＤＤ）４、モデム５およびキーボードコントローラ６を備えている。

【００１１】ＣＰＵ１は、このコンピュータシステム全体の制御を司るものであり、システムメモリ２に格納されたオペレーティングシステムやハードウェアドライバ、ユーティリティを含むアプリケーションプログラムなどを実行制御する。

【００１２】システムメモリ２は、このコンピュータシステムの主記憶装置となるメモリデバイスであり、ＣＰＵ１によって実行制御されるオペレーティングシステムやハードウェアドライバ、ユーティリティを含むアプリケーションプログラム、およびこれらの実行に用いられる各種処理データなどを格納する。

【００１３】ディスプレイコントローラ３は、ユーザインタフェースのアウトプットを司るデバイスであり、ＣＰＵ１により描画されたビデオメモリ（ＶＲＡＭ）上の表示データをＬＣＤなどのディスプレイに表示する。

【００１４】磁気ディスク装置（ＨＤＤ）４は、このコンピュータシステムの補助記憶装置となる大容量のメモリデバイスであり、システムメモリ２にロードされるプログラムや処理データ、あるいはシステムメモリ２から退避されるプログラムや処理データなどを格納する。

【００１５】モデム５は、回線網（アナログ）との接続に用いられるデバイスであり、回線網を介在させて他のコンピュータシステムとの間で送受信されるデータをアナログ／デジタル変換する。

【００１６】そして、キーボードコントローラ６は、ユーザインタフェースのインプットを司るデバイスであ

り、キーボードやマウスから送られる操作内容を示す制御信号を受け取ってＣＰＵ１に転送する。

【００１７】図２には、このような機器構成をもつこの実施形態のコンピュータシステムの機能ブロックが示されている。なお、この図２では、米国マイクロソフト社により開発されたＷｉｎｄｏｗｓ９５オペレーティングシステム下での機能ブロックを示すが、この発明は、これに限定されるものではない。すなわち、オペレーティングシステムの形式に依存して変形させることは、この発明の範囲から逸脱することなしに容易に可能である。

【００１８】この実施形態のコンピュータシステムでは、モデム５へのアクセスを必要とする一つまたはそれ以上の通信アプリケーションプログラム１１（リモートネットワークアクセス、ファックスアプリケーション、電子メールアプリケーションおよび掲示板サービスアプリケーションなど）に対するサポートを提供する。この通信アプリケーションプログラム１１は、モデム５へのアクセスを必要とするものであり、そのモデム５に対するＩ／Ｏコマンドの発行によりデータＩ／Ｏを実行する。

【００１９】通信ドライバ１２は、ＣＯＭＭ　ＡＰＩ１２１をサポートし、通信アプリケーションプログラム１１に装置に依存しない方法でモデム５を使用することを許可する。この通信ドライバ１２は、通信アプリケーションプログラム１１が通信デバイスをオープンし、通信デバイスから読み出し、通信デバイスに書込むことを許可する機能を含む。また、ＣＯＭＭ　ＡＰＩ１２１のデータＩ／Ｏ機能を呼び出し、最終的にモデム５を通じてデータを送受信する。そして、ユーザは、通信アプリケーションプログラム１１からユーザインタフェース機能を通じてモデム５の動作特性選択し、コンフィギュレーション機能を通じてモデム５の動作特性を決定または変更することができる。

【００２０】ＶＣＯＭＭ１３は、いわゆる仮想デバイスであり、システムにおけるシリアルポート、パラレルポートおよびモデムなどといった通信資源へのすべてのアクセスを管理する。そして、通信ドライバ１２は、このＶＣＯＭＭ１３によって供給されるＶＣＯＭＭサービス１３１によってＶＣＯＭＭ１３と通信する。

【００２１】ポートドライバ１４は、デバイス特定のコードを含んでおり、これにより通信アプリケーションプログラム１１がポートに接続された特定のデバイスと通信することを可能にする。

【００２２】そして、この実施形態のコンピュータシステムは、ポートモニタドライバ１７と表示アプリケーションプログラム１６とを備えた点を特徴とするものであり、ポートモニタドライバ１７は、ＶＣＯＭＭ１３とポートドライバ１４との間の通信の監視、通信データの解析および通信データの変更などを可能とするものである。なお、図２は、通信アプリケーションプログラム１１とモデム５との間の通信の監視や解析、変更を施すための一例を示すものであり、この発明はこの構成に限定されるものではない。たとえば、ポートモニタドライバ１７は、通信ドライバ１２とＶＣＯＭＭ１３との間の通信を監視、解析または変更することも可能である。

【００２３】従来、通信ドライバ１２は、ＶＣＯＭＭ１３によって供給されるＶＣＯＭＭサービス１３１を用いてＶＣＯＭＭ１３と通信し、また、ＶＣＯＭＭ１３は、ポートドライバ１４によって供給されるポートドライバ関数１４１を用いてポートドライバ１４と通信するが、図２に示すこの実施形態においては、ポートモニタドライバ１７が通信ドライバ１２のＶＣＯＭＭサービス１３１の利用をフック（横取り、あるいは盗み取り）したり、また、ＶＣＯＭＭ１３のポートドライバ関数１４１の利用をフックすることにより、実質的に、通信ドライバ１２はポートモニタドライバ１７と通信し、また、ポートモニタドライバ１７はＶＣＯＭＭ１２と通信することになり、さらに、ＶＣＯＭＭ１３はポートモニタドライバ１７と通信し、また、ポートモニタドライバ１７はポートドライバ１４と通信することになる。

【００２４】そして、表示アプリケーションプログラム１６は、ポートモニタドライバ１７が通信ドライバ１２のＶＣＯＭＭサービス１３１の利用やＶＣＯＭＭ１３のポートドライバ関数１４１の利用をフックすることによって得られる各種データ通信情報（回線の切断理由、接続先電話番号および課金情報など）をポップアップ画面などによりユーザに対して表示するものである。

【００２５】図３は、この実施形態のコンピュータシステムにおけるシステムメモリ２上の構成例を示すものである。ＶＣＯＭＭ１３は、ＶＣＯＭＭサービス１３１のエントリをシステムメモリ２上に確保している。図３には、＿ＶＣＯＭＭ＿ＯｐｅｎＣｏｍｍ１３１１、＿ＶＣＯＭＭ＿Ａｄｄ＿Ｐｏｒｔ１３１２、＿ＶＣＯＭＭ＿ＣｌｏｓｅＣｏｍｍ３１３などのＶＣＯＭＭサービス１３１の例が示されている。通信ドライバ１２は、通信アプリケーションプログラム１１からのＣＯＭＭ　ＡＰＩ１２１を利用した命令に基づき、＿ＶＣＯＭＭ＿ＯｐｅｎＣｏｍｍ１３１１、＿ＶＣＯＭＭ＿Ａｄｄ＿Ｐｏｒｔ１３１２などのＶＣＯＭＭサービス１３１を利用して、通信ポートのオープンのための準備やポートの管理機構の構築などを行ない、最後に＿ＶＣＯＭＭ＿ＣｌｏｓｅＣｏｍｍ１３１３を用いてポートをクローズする。

【００２６】ポートドライバ１４は、オープンされたポートを管理するためのＰｏｒｔＤａｔａ構造体１４１、ポートをオープンするためのＰｏｒｔＯｐｅｎ関数ポインタ１４２などをメモリ上に確保している。また、ＰｏｒｔＤａｔａ構造体１４１には、構造体のサイズ情報１４１１、構造体のバージョン情報１４１２、ポートＩ／Ｏを行なうための様々な関数へのポインタ群１４３へのポインタ１４１３、ポインタ群１４３に含まれる関数の

数情報１４１４などが要素として含まれ、ポートＩ／Ｏ関数ポインタ群１４３には、ポートをクローズするための関数ＰｏｒｔＣｌｏｓｅへのポインタ１４３１、ポートからデータを読み出すための関数ＰｏｒｔＲｅａｄへのポインタ１４３２、ポートへデータを書き出すための関数ＰｏｒｔＷｒｉｔｅへのポインタ１４３３などが含まれる。

【００２７】ポートモニタドライバ１７は、ポートをオープンするための関数であるＰｏｒｔＯｐｅｎへのポインタ１４２をコピーして格納する領域ＯｐｅｎＦｕｎｃ１７１１、ポートＩ／Ｏ関数ポインタ群１４３へのポインタをコピーして格納するＯｒｉｇｉｎａｌＦｕｎｃ１７１２、ポートモニタドライバ１７独自のポートＩ／Ｏ関数ポインタ群１７２へのポインタを格納するＬｏｃａｌＦｕｎｃ１７１３などを持った構造体ＯｐｅｎＩｎｆｏ１７１を含んでいる。ポートモニタドライバ１７独自のポートＩ／Ｏ関数群１７３には、たとえばポートをオープンするための準備を行なうＬｏｃａｌＡｄｄＰｏｒｔ１７３１、ポートをオープンするＬｏｃａｌＯｐｅｎ１７３２、ポートをクローズするＬｏｃａｌＣｌｏｓｅ１７３３、ポートからデータを読み出すＬｏｃａｌＲｅａｄ１７３４、ポートへデータを書き出すＬｏｃａｌＷｒｉｔｅ１７３５などがある。また、ポートモニタドライバ１７独自のポートＩ／Ｏ関数ポインタ群１７２には、ＬｏｃａｌＣｌｏｓｅ１７３３へのポインタＬｏｃａｌＰｏｒｔＣｌｏｓｅ１７２１、ＬｏｃａｌＲｅａｄ１７３４へのポインタＬｏｃａｌＰｏｒｔＲｅａｄ１７２２、ＬｏｃａｌＷｒｉｔｅ１７３５へのポインタＬｏｃａｌＰｏｒｔＷｒｉｔｅ１７２３などが含まれている。

【００２８】また、ライトバッファ１７４およびリードバッファ１７５は、それぞれポートへ書き出すデータを一時格納するバッファ、ポートから読み出したデータを一時格納するバッファである。

【００２９】さらに、ポートモニタドライバ１７は、表示アプリケーションプログラム１６に情報を通知するため、表示アプリケーションプログラム１６が持つ関数へのポインタ群ＣａｌｌｂａｃｋＦｕｎｃ構造体１７６を含んでいる。図３は、表示アプリケーションプログラム１７６が持つ３種類の関数へのポインタをＣａｌｌｂａｃｋＦｕｎｃ構造体１７６が含んでいる例であり、ＣａｌｌｂａｃｋＦｕｎｃ１（１７６１）、ＣａｌｌｂａｃｋＦｕｎｃ２（１７６２）、ＣａｌｌｂａｃｋＦｕｎｃ３（１７６３）がそれである。

【００３０】以下、図４乃至図１２のフローチャートを参照してこのコンピュータシステムの動作原理を説明する。図４は、ポートモニタドライバ１７の初期化を示すフローチャートである。たとえば表示アプリケーションプログラム１６などの上位アプリケーションによりポートモニタドライバ１７がロードされると、ポートモニタドライバ１７は、最初に初期化作業として、ＶＣＯＭＭサービス１３１のうちの＿ＶＣＯＭＭ＿Ａｄｄ＿Ｐｏｒｔ１３１２のフックを行なう（ステップＡ１）。従来、通信ドライバ１２がＶＣＯＭＭサービス１３１のうちの＿ＶＣＯＭＭ＿Ａｄｄ＿Ｐｏｒｔ１３１２をコールした際には、ＶＣＯＭＭ１３内に用意されている関数がコールされるが、その関数へのポインタが登録されている部分をポートモニタドライバ１７内のＬｏｃａｌＡｄｄＰｏｒｔ１７３１と置き換えることにより、ＬｏｃａｌＡｄｄＰｏｒｔ１７３１で示されるポートモニタドライバ１７内のローカル関数がコールされるようになる。そして、ポートモニタドライバ１７は、そのローカル関数内で独自の処理を行なった後、元の本来の関数をコールし直すことにより、フックしたこととなる。次に、ポートモニタドライバ１７は、ＯｐｅｎＩｎｆｏ構造体１７１の初期化を行なう（ステップＡ２）。これは、たとえば構造体の各要素に０（ゼロ）を代入して初期化するといった、一般的なメモリ初期化作業である。

【００３１】図５は、ＣａｌｌｂａｃｋＦｕｎｃ構造体１７６へのコールバック関数の登録を示すフローチャートである。たとえば表示アプリケーションプログラム１６などの上位アプリケーションによりポートモニタドライバ１７がロードされる際、表示アプリケーションプログラム１６は、自身に含まれる関数のポインタをポートモニタドライバ１７へ通知することにより、その関数のポインタをＣａｌｌｂａｃｋＦｕｎｃ構造体１７６へ登録する（ステップＢ１）。この処理により、ポートモニタドライバ１７は、情報を表示アプリケーションプログラム１６に通知するためにどの関数（ルーチン）をコールすれば良いか知ることができる。

【００３２】図６は、通信ドライバ１２がＶＣＯＭＭサービス１３１のうちの＿ＶＣＯＭＭ＿Ａｄｄ＿Ｐｏｒｔ１３１２をコールした際、図４のステップＡ１の処理を行なったことによりポートモニタドライバ１７内のローカル関数ＬｏｃａｌＡｄｄＰｏｒｔ１７３１が呼ばれたときに、そのローカル関数ＬｏｃａｌＡｄｄＰｏｒｔ１７３１が実際に実行する処理を示すフローチャートである。＿ＶＣＯＭＭ＿Ａｄｄ＿Ｐｏｒｔコールにより呼ばれるこの関数は、この実施形態のＷｉｎｄｏｗｓ９５のオペレーティングシステムの場合、　ポートを実際にオープンする際にコールされる関数のポインタ（この場合はＰｏｒｔＯｐｅｎ１４２）を得られる仕様となっているため、このＬｏｃａｌＡｄｄＰｏｒｔ１７３においても同様にＰｏｒｔＯｐｅｎ１４２を得て、それをＯｐｅｎＩｎｆｏ構造体１７１のＯｐｅｎＦｕｎｃ１７１１へ格納する（ステップＣ１）。これにより、ＯｐｅｎＦｕｎｃ１７１１ポインタは、ＰｏｒｔＯｐｅｎ１４２を指し示すこととなる（矢印ａ）。次に、この関数は、フックしなかった場合に＿ＶＣＯＭＭ＿Ａｄｄ＿Ｐｏｒｔコールによりコールされるはずであった本来の関数をコー

ルする（ステップＣ２）。このとき、ポートモニタドライバ１７が独自に持つオープン関数へのポインタＬｏｃａｌＯｐｅｎ１７３２をアーギュメントとして渡すことにより、それ以後、ポートを実際にオープンする際には、本来の関数ＰｏｒｔＯｐｅｎ１４２がコールされる代わりに、ローカル関数ＬｏｃａｌＯｐｅｎ１７３２がコールされることになる。

【００３３】図７は、図６のステップＣ２の処理を行なったことにより、ポートをオープンする際にポートモニタドライバ１７内のローカル関数ＬｏｃａｌＯｐｅｎ１７３２がコールされたときに、そのローカル関数ＬｏｃａｌＯｐｅｎ１７３２が実際に実行する処理を示すフローチャートである。この関数は、まず始めに、図６のステップＣ１の処理によりＯｐｅｎＦｕｎｃ１７１１に格納した本来のＰｏｒｔＯｐｅｎ１４２をコールする（ステップＤ１）。Ｗｉｎｄｏｗｓ９５のオペレーティングシステムにおいては、本来のＰｏｒｔＯｐｅｎ１４２を返り値としてＰｏｒｔＤａｔａ構造体１４１へのポインタが得られるので、その要素よりＰｏｒｔＦｕｎｃｔｉｏｎｓ１４１３を取得し、ポートＩ／Ｏ関数ポインタ群１４３へのポインタを得て、それをＯｒｉｇｉｎａｌＦｕｎｃ１７１２へ格納する（ステップＤ２）。さらに、ポートモニタドライバ１７が持つ独自のポートＩ／Ｏ関数ポインタ群１７２内に、本来のポートＩ／Ｏ関数ポインタ群１４３の内容をコピーした後（ステップＤ３）、ポートモニタドライバ１７にてフックする機能（関数）のみ、独自のポートＩ／Ｏ関数ポインタ群１７２の内容を変更する（ステップＤ４）。図３ではは、一例として、ポートをクローズする際にコールされるＰｏｒｔＣｌｏｓｅ１４３１、ポートからデータを読み出すためのＰｏｒｔＲｅａｄ１４３２、ポートへデータを書き出すためのＰｏｒｔＷｒｉｔｅ１４３３をフックするために、独自のポートＩ／Ｏ関数ポインタ群１７２の要素のうち、それら三つの関数のポインタをそれぞれＬｏｃａｌＰｏｒｔＣｌｏｓｅ１７２１、ＬｏｃａｌＰｏｒｔＲｅａｄ１７２２、ＬｏｃａｌＰｏｒｔＷｒｉｔｅ１７２３に変更している。ＬｏｃａｌＰｏｒｔＣｌｏｓｅ１７２１は、ポートモニタドライバ１７が独自に持つ関数ＬｏｃａｌＣｌｏｓｅ１７３３、ＬｏｃａｌＰｏｒｔＲｅａｄ１７２２は、ポートモニタドライバ１７が独自に持つ関数ＬｏｃａｌＲｅａｄ１７３４、ＬｏｃａｌＰｏｒｔＷｒｉｔｅ１７２３は、ポートモニタドライバ１７が独自に持つ関数ＬｏｃａｌＷｒｉｔｅ１７３５を指し示す。そして、すでに得ているＰｏｒｔＤａｔａ構造体１４１に含まれるＰｏｒｔＦｕｎｃｔｉｏｎｓ１４１３をポートモニタドライバ１７独自のポートＩ／Ｏ関数ポインタ群１７２へのポインタと置き換える（ステップＤ５）。こうすることにより、それ以降、ＶＣＯＭＭ１３がオペレーティングシステムの動作に基づいてポートドライバ１４に含まれる関数をコールする際、コールする関数へのポインタを含んでいるＰｏｒｔＤａｔａ構造体１４１のＰｏｒｔＦｕｎｃｔｉｏｎｓ１４１３がポートモニタドライバ１７所有のポートＩ／Ｏ関数ポインタ群１７１を指しているため、ポートモニタドライバ１７内のローカル関数ＬｏｃａｌＣｌｏｓｅ１７３３、ＬｏｃａｌＲｅａｄ１７３４およびＬｏｃａｌＷｒｉｔｅ１７３５などがコールされることになる。

【００３４】図８は、ポートモニタドライバ１７内のローカル関数ＬｏｃａｌＲｅａｄ１７３４がコールされた際の処理を示すフローチャートである。通信アプリケーションプログラム１１からポートリード要求があった際、図３乃至図７に示した処理によりＰｏｒｔＲｅａｄ１４３２のコールがフックされ、ＬｏｃａｌＲｅａｄ１７３４がコールされる。この関数では、まず始めに、ＯｐｅｎＩｎｆｏ構造体１７１に含まれるＯｒｉｇｉｎａｌＦｕｎｃ１７１２より、本来のポートＩ／Ｏ関数ポインタ群１４３を得て、さらにそこから得られるＰｏｒｔＲｅａｄ１４３２をコールすることにより、本来のポートリード処理を行なう（ステップＥ１）。そして、ポートリード処理より得られた、たとえばモデムの出力などのリードデータを解析し（ステップＥ２）、電話通信切断時の回線切断理由に関わる情報や電話通信終了時の課金に関わる情報などを取得する。

【００３５】図９は、ポートモニタドライバ１７内のローカル関数ＬｏｃａｌＷｒｉｔｅ１７３５がコールされた際の処理を示すフローチャートである。通信アプリケーションプログラム１１からポートライト要求があった際、図３乃至図7 に示した処理によりＰｏｒｔＷｒｉｔｅ１４３３のコールがフックされ、ＬｏｃａｌＷｒｉｔｅ１７３５がコールされる。この関数では、まず始めに、ＯｐｅｎＩｎｆｏ構造体１７１に含まれるＯｒｉｇｉｎａｌＦｕｎｃ１７１３より、本来のポートＩ／Ｏ関数ポインタ群１４３を得て、さらにそこから得られるＰｏｒｔＷｒｉｔｅ１４３３をコールすることにより、本来のポートライト処理を行なう（ステップＦ１）。そして、ポートライト要求から得られた、たとえば通信アプリケーションプログラム１１からモデムへのＡＴコマンド書き出しなどのライトデータを解析し（ステップＦ２）、接続先の電話番号などの情報を取得する。

【００３６】図１０は、ポートモニタドライバ１７内のローカル関数ＬｏｃａｌＣｌｏｓｅ１７３３がコールされた際の処理を示すフローチャートである。通信アプリケーションプログラム１１からポートクローズ要求があった際、図３乃至図６に示した処理によりＰｏｒｔＣｌｏｓｅ１４３１のコールがフックされ、ＬｏｃａｌＣｌｏｓｅ１７３３がコールされる。この関数では、まず始めに、ＯｐｅｎＩｎｆｏ構造体１７１に含まれるＯｒｉｇｉｎａｌＦｕｎｃ１７１３より、本来のポートＩ／Ｏ関数ポインタ群１４３を得て、さらにそこから得られるＰｏｒｔＣｌｏｓｅ１４３１をコールすることにより、

本来のポートクローズ処理を行なう（ステップＧ１）。そして、ＯｐｅｎＩｎｆｏ構造体１７１を初期化する（ステップＧ２）。

【００３７】図１１は、図８のステップＥ２のリードデータ解析処理の詳細を示すフローチャートである。図８のステップＥ１の処理によりポートからのリードデータが得られた際、まず、リードデータをリードバッファ１７５にコピーする（ステップＨ１）。たとえば通信アプリケーションプログラム１１とモデム５との間でＩ／ＯされるＡＴコマンドのリザルトコード文字列を解析する場合には、ＣＲ（キャリッジリターン）コートがひとつのリザルトコードの区切りになるので、リードバッファ１７５にはＣＲコードまでを一時格納する。そして、たとえば通信が終了または切断された場合にモデム５から出力されるリザルトコード「ＮＯ　ＣＡＲＲＩＥＲ」を検出するため、リードバッファ１７５に格納された文字列のうちの最初の１０文字を「ＮＯ　ＣＡＲＲＩＥＲ」文字列と比較し（ステップＨ２）、一致した場合には（ステップＨ２の“一致”）、ＣａｌｌｂａｃｋＦｕｎｃ構造体１７６に格納される任意の関数、たとえばＣａｌｌｂａｃｋ　Ｆｕｎｃ１をコールし、リードバッファ１７５の内容を表示アプリケーションプログラム１６に通知する（ステップＨ３）。一方、表示アプリケーションプログラム１６は、たとえば得られたリードバッファ１７５の文字列に「ＮＯ　ＣＡＲＲＩＥＲ：ＲＥＣＥＩＶＥＤ　ＲＥＦＵＳＥＤ」が格納されている場合には、「：ＲＥＣＥＩＶＥＤ」文字列以降が回線切断の理由として、「ＲＥＦＵＳＥＤ」を切り出し、図１３に示すように画面上に回線切断理由を表示する。この例は、電話相手先が着信拒否をしたために回線が切断されたことを意味している。

【００３８】また、たとえば課金情報を表示する場合には、通信が終了した場合にモデム５から出力されるリザルトコード「ＣＨＡＲＧＥ＝」を検出してＣａｌｌｂａｃｋＦｕｎｃ構造体１７６に格納される任意の関数をコールすることにより、リードバッファ１７５の内容を表示アプリケーションプログラム１６に通知する。この場合の文字列は、たとえば通信料金が１００円のときは「ＣＨＡＲＧＥ＝１００」、通信料金が１２７０円のときは「ＣＨＡＲＧＥ＝１２７０」のように構成されるので、表示アプリケーションプログラム１６は、「１００」や「１２７０」を切り出して、課金情報として画面上に表示すればよい。

【００３９】図１２は、図９のステップＦ２のライトデータ解析処理の詳細を示すフローチャートである。図９のステップｆ１の処理により通信アプリケーション１１からモデム５へのライトデータが得られた際、まず、ライトデータをライトバッファ１７４にコピーする（ステップＩ１）。たとえば通信アプリケーションプログラム１１とモデム５との間でＩ／ＯされるＡＴコマンドを解析する場合には、ＣＲ（キャリッジリターン）コードがひとつのＡＴコマンドの区切りになるので、ライトバッファ１７４にはＣＲコードまでを一時格納する。そして、たとえば通信アプリケーションプログラム１１がＡＴコマンドを用いて電話発呼処理を行なう際の接続先電話番号を得るべく「ＡＴＤ」（発呼を行うためのＡＴコマンド）という３文字の文字列を検出するため、ライトバッファ１７４に格納された文字列のうちの最初の３文字を「ＡＴＤ」文字列と比較し（ステップＩ２）、一致した場合には（ステップＩ２の“一致”）、たとえば「ＡＴＤ０３－７７７－７７７７」のようにＡＴコマンド文字列に接続先電話番号が続いているので、ＡＴＤ以降の数字および括弧やハイフンなどの文字列を一時的なバッファに格納する（ステップＩ３～ステップＩ４）。そして、一連の電話番号が得られてから、ＣａｌｌｂａｃｋＦｕｎｃ構造体１７６に格納される任意の関数、たとえばＣａｌｌｂａｃｋ　Ｆｕｎｃ２をコールし、ライトバッファ１７４の内容を表示アプリケーションプログラム１６に通知する（ステップＩ５）。一方、表示アプリケーションプログラム１６は、たとえば得られた接続先電話番号の情報を図１３のように画面上に表示する。

【００４０】このように、この実施形態のコンピュータシステムにおいては、通信アプリケーションプログラム１１がモデム５の出力する切断理由、接続先電話番号および課金情報などの各種データ通信情報を無視するものであっても、この通信アプリケーションプログラム１１とは別途独立して、すなわち、この通信アプリケーションプログラム１１の改良を伴なうことなく、ユーザにとって有意義な情報を表示することが可能となる。

【００４１】

【発明の効果】以上詳述したように、この発明においては、コンピュータシステム上の任意のデータ通信アプリケーションプログラムがモデムなどを介してデータ通信を行なう際に、そのデータ通信アプリケーションプログラムからモデムに対して発行されるリードやライトなどのＩ／Ｏコマンドを盗み取り、この盗み取ったＩ／Ｏコマンドから回線の切断理由、接続先電話番号および課金情報などの各種データ通信情報を抽出するとともに、その盗み取ったＩ／Ｏコマンドを本来の受け取り先に何もなかったかのように返却するため、通信アプリケーションプログラムの改良などを伴なわずに、この抽出した各種データ通信情報をユーザに知らせることが可能となる。

【図面の簡単な説明】

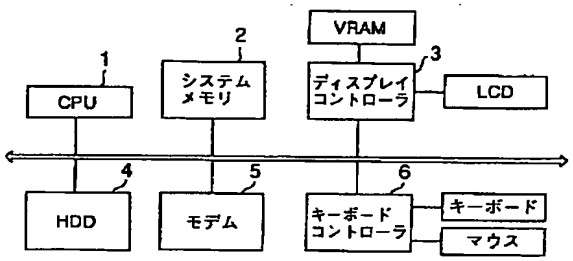
【図１】この発明の実施形態に係るコンピュータシステムの概略機器構成を示す図。

【図２】同実施形態のコンピュータシステムの機能ブロックを示す図。

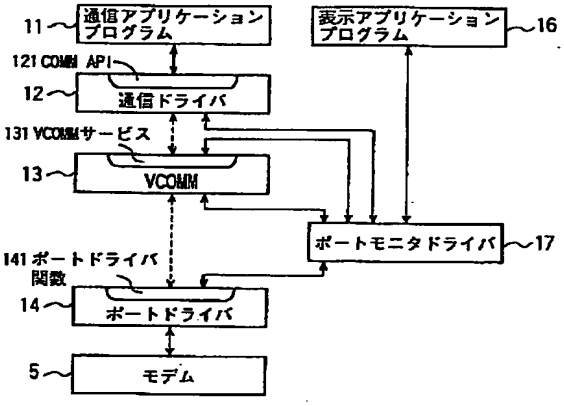
【図３】同実施形態のコンピュータシステムにおけるシステムメモリ上の構成例を示す図。

【図４】同実施形態のポートモニタドライバの初期化を示すフローチャート。
【図５】同実施形態のＣａｌｌｂａｃｋＦｕｎｃ構造体へのコールバック関数の登録を示すフローチャート。
【図６】同実施形態の通信ドライバがＶＣＯＭＭサービスのうちの_ＶＣＯＭＭ_Ａｄｄ_Ｐｏｒｔ１をコールした際に呼ばれるポートモニタドライバ内のローカル関数ＬｏｃａｌＡｄｄＰｏｒｔが実行する処理を示すフローチャート。
【図７】同実施形態のポートをオープンする際にコールされるポートモニタドライバ内のローカル関数ＬｏｃａｌＯｐｅｎが実行する処理を示すフローチャート。
【図８】同実施形態のポートモニタドライバ内のローカル関数ＬｏｃａｌＲｅａｄがコールされた際の処理を示すフローチャート。
【図９】同実施形態のポートモニタドライバ内のローカル関数ＬｏｃａｌＷｒｉｔｅがコールされた際の処理を示すフローチャート。
【図１０】同実施形態のポートモニタドライバ内のローカル関数ＬｏｃａｌＣｌｏｓｅがコールされた際の処理を示すフローチャート。
【図１１】同実施形態のポートモニタドライバ内のローカル関数ＬｏｃａｌＲｅａｄ中のリードデータ解析処理の詳細を示すフローチャート。
【図１２】同実施形態のポートモニタドライバ内のローカル関数ＬｏｃａｌＷｒｉｔｅ中のライトデータ解析処理の詳細を示すフローチャート。
【図１３】同実施形態の表示アプリケーションプログラムによる各種データ通信情報の表示例を示す図。
【符号の説明】
１…ＣＰＵ、２…システムメモリ、３…ディスプレイコントローラ、４…磁気ディスク装置（ＨＤＤ）、５…モデム、６…キーボードコントローラ、１１…通信アプリケーションプログラム、１２…通信ドライバ、１３…ＶＣＯＭＭ、１４…ポートドライバ、１６…表示アプリケーションプログラム、１７…ポートモニタドライバ。

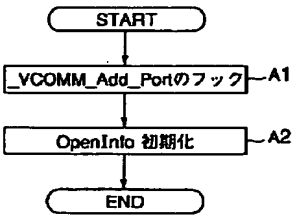
【図１】



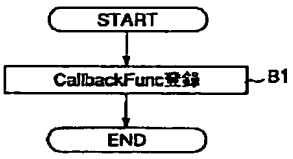
【図２】



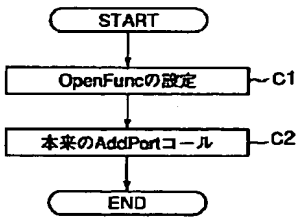
【図４】



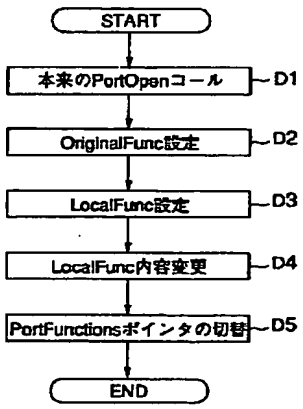
【図５】



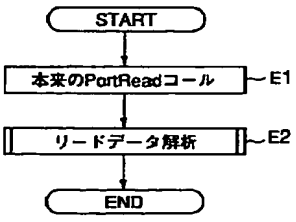
【図６】



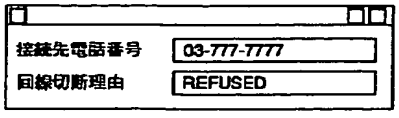
【図７】



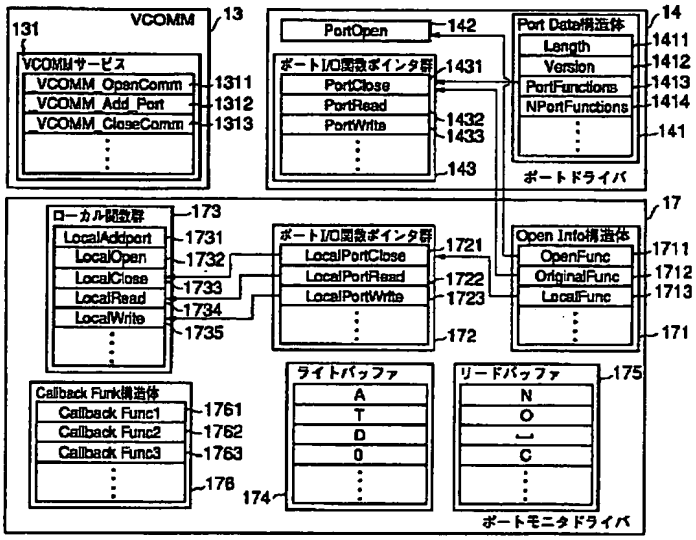
【図８】



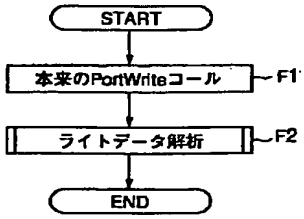
【図１３】



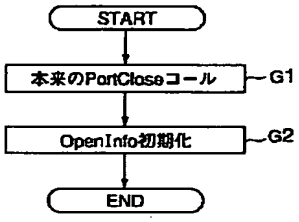
【図３】



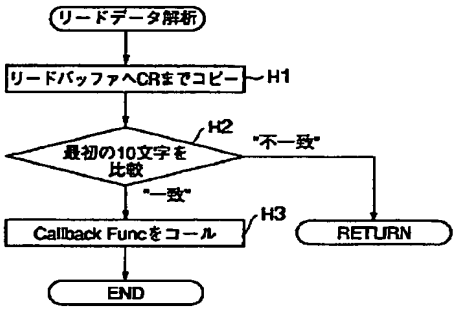
【図９】



【図１０】



【図１１】



【図１２】